Optimization of Transmission Control Protocol and Feedback Control Mechanisms for Wireless Internet

A thesis submitted for the Degree of Master of Science in Computing (Thesis)

by

Phuong N. Nhan

Faculty of Information Technology

UNIVERSITY OF TECHNOLOGY SYDNEY

July, 2004

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis had not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis

Signature of Candidate

Preface

This research is funded by SPIRT (Strategic Partnership with Industry – Research and Training) Scheme in conjunction with the Motorola Australian Research Centre. The objective of the research project is to explore, propose, design and implement several algorithms that are applicable to the wireless networks in order to solve outstanding problems.

Firstly, the research investigates the relationship between packet loss and network congestion and introduces a feedback based end-to-end congestion control algorithm to the wireless network. The next algorithm is the new design of Explicit Loss Notification (ELN) at Base station in Wired-Cum-Wireless networks. With the combination of new ELN algorithm and Wireless FICC algorithm, the end-to-end performance and fairness are greatly improved by eliminating the misinterpretation of error related lost packets from congestion. Finally, the research investigates the effects of network congestion, which often happens over low bandwidth wireless link, and QoS performance (e.g. fairness, delay variation) of multiple sessions of TCP traffic in a hybrid network. We propose a framework, which consists of two main algorithms, feedback based congestion control and Explicit Window Adaptation (EWA).

Abstract

All current versions of reliable Transmission Control Protocol (TCP) react to packet losses differently and adjust the TCP congestion window in various ways. These protocols assume congestion in the network to be the primary cause for packet losses and unusual delays. TCP performs well over wired networks by adapting to end-to-end delays and packet losses caused by congestion. The TCP sender uses the cumulative acknowledgements it receives to determine which packets have reached the receiver, and provides reliability by retransmitting lost packets. The sender identifies the loss of a packet either by the arrival of several duplicate cumulative acknowledgements (say, three ACKs) or the absence of an acknowledgement for the packet within a timeout. TCP reacts to packet losses by reducing its transmission (congestion) window size before retransmitting packets, initiating congestion window or avoidance mechanisms and backing off its retransmission timer. These measures result in a reduction in the load on the intermediate links, thereby controlling the congestion in the network. Unfortunately, when packets are lost in the networks for reasons other than congestion, these measures result in an unnecessary reduction in end-to-end throughput and sub-optimal performance.

Wireless links typically have much higher bit error rates. This implies that packet loss would occur frequently. If no error correction is attempted at lower layer, TCP will exercise its congestion control procedure unnecessarily and the throughput will be reduced significantly. If the link layer performs error control by performing the retransmission itself, packet transmission time will vary greatly, sometime even exceeding TCP retransmission time out and again TCP slow start will occur. In wireless networks, "packet loss" problem is also encountered during handover when a mobile device moves from the coverage of one cell to that of another. During the handover, if the mobile station decides to make a handover before the segments are transmitted over the air interface, it is likely that some TCP segments buffered in a base station may be forwarded to another base station. This results in excessive segment delay or loss.

Thus, there is a clear demand for methods that can suppress the problems caused by the wireless environment. Recently, several techniques have been developed to improve end-to-

end TCP performance over wireless links. They can be classified into three categories: end-to-end TCP, split TCP and link layer TCP. However, they have not addressed these problems successfully.

In this thesis, we propose, design and implement several algorithms that are applicable to the wireless networks in order to solve outstanding problems. Firstly, the research investigates the relationship between packet loss and network congestion and introduces a feedback based end-to-end congestion control algorithm to the wireless network. This algorithm is a modification of a Fair Intelligent Congestion Control (FICC) proposed in [19]. The innovation of the algorithm is to modify the original FICC in such a way that the queue lengths can be effectively controlled when it is jointly employed with TCP in the wireless network.

The next algorithm is the new design of Explicit Loss Notification (ELN) at base station in Wired-Cum-Wireless networks. With the combination of new ELN algorithm and Wireless FICC algorithm, the end-to-end performance and fairness are greatly improved by eliminating the misinterpretation of error related lost packets from congestion.

Finally, the research investigates the effects of network congestion, which often happens over low bandwidth wireless link, and QoS performance (e.g. fairness, delay variation) of multiple sessions of TCP traffic in a hybrid network. We propose a framework, which consists of two main algorithms, feedback based congestion control and Explicit Window Adaptation (EWA).

Acknowledgements

This research was started when I was at School of Information Technologies (formerly Basser Department of Computer Science), the University of Sydney. It was resumed when I moved to the Department of Computer Systems, Faculty of Information Technology University of Technology, Sydney (UTS).

Many people have helped me all the way during the development of this thesis. First and foremost, I would like to thank my main supervisor A/Prof. Doan Hoang who took me on as his student, provided a motivating, enthusiastic, and critical atmosphere during the many discussions we had. It was a great pleasure for me to conduct this thesis under his supervision.

I am grateful to my co-supervisor Dr Vinod Mirchandani at Motorola Australian Research Centre, Sydney. He gave me very supportive guidelines in order to achieve the target of the research. We had meetings and discussions on my research on regular basis at Motorola Australian Research Centre, Sydney since I started my research.

I benefited greatly from Dr Hong Zhou providing constructive comments during my thesis time as well as giving valuable detailed comments on simulation results analysis. She also gave me great help in C++ coding for ns-2 implementations and excellent suggestions on algorithm designs.

I was fortunate to be a member of the Advance Research on Networking (ARN) group. I would like to thank everyone in the group including Xiaomei, Qing, Ian, Chi, Bushar, Hanh and Joe who shared experiences and knowledge with me from my starting points; a special thank to Ming Li who helped me on gaining general networking knowledge as well as on Network Simulation (ns-2).

Last but not least, I owe a special debt of gratitude to my grandparents and parents. They have, more than anyone else, been the reason for me to get this far. They give me love, and emotions to overcome all difficulties in life.

To my grandparents

and my parents

Table of Contents

Preface iii

Abstract iv

Acknowledgements vi

Table of Contents viii

List of Figures xii

List of Tables xiv

List of Abbreviations and Terminology xv

Chapter 1. Introduction **1**

1.1 Motivation 1

1.2 Contributions of the Thesis 5

1.2.1 Wireless Fair Intelligent Congestion Control (WFICC) 5
1.2.2 Wireless Fair Intelligent Congestion Control with Explicit Loss Notification (WFICC+ELN) 6
1.2.3 Feedback-Based Congestion Control and Explicit Window Adaptation in a Hybrid Network with Wired and Wireless Links ... 7

1.3 Structure of the Thesis 9

Chapter 2. Background and Related Work **11**

2.1 The Internet Protocol Architecture 11
2.1.1 Heterogeneity 12
2.1.2 Reliability 12
2.1.3 Incremental Deployment 13

2.2 Transmission Control Protocol (TCP) 14
2.2.1 Cumulative Acknowledgement 15
2.2.2 Loss Recovery 15
2.2.3 Congestion Avoidance and Control 17
2.2.4 Connection Management 18
2.2.5 Window-based vs. Rate-based Congestion Control 18

2.3 End-to-end TCP Enhancements 19
2.3.1 Selective Acknowledgement (SACK) 19
2.3.2 Newreno 20
2.3.3 Explicit Congestion Notification 20
2.3.4 Explicit Loss Notification 21

2.4 Wireless Networks 21
2.4.1 Wireless Technology Overview 22
2.4.2 IEEE 802.11x Concepts

2.5 Wireless Bit-errors and User Mobility 25
2.5.1 Reasons for Wireless Data Corruption 26
2.5.2 End-to-end Protocols 27
2.5.3 Split-Connection Protocols 28
2.5.4 Link-layer Protocols

2.6 Summary 29

Chapter 3. Common Tools 30

3.1 Overview of Fair Intelligent Congestion Control 30

3.2 Fair Intelligent Congestion Control Deployment over Wireless Links 33

3.3 Overview of Network Simulation (NS-2) 35
3.3.1 Overview 35
3.3.2 Network Components 39
3.3.3 Structure of Network Simulation 2 (NS-2) 44

3.4 General Simulation Setup 46

3.5 Summary 49

Chapter 4. Wireless Fair Intelligent Congestion Control (WFICC) 50

4.1 Introduction 50

4.2. Operations 51

4.2.1 At Sources 51
4.2.2 At Routers 52

4.3. Simulation Results and Analysis 52

4.3.1 WFICC on Wired Networks 52
- Queue Length 55
- Congestion Window 57
- Goodput 58

4.3.2. WFICC ON Wireless Networks 59

4.3.2.1 No Errors 61
- Queue Length 61
- Congestion Window 62
- Goodput 63

4.3.2.2 With Errors 64
- Queue Length 65
- Congestion Window 66
- Goodput 67

4.4 Comparisons and Evaluations 68

4.4.1 Queue Length and Packet Lost 68
4.4.2 Delay 69
4.4.3 Goodput 70
4.4.4 Fairness 71

4.5 Summary 72

Chapter 5. Window-Based Fair Intelligent Congestion Control Algorithm (WFICC) with Explicit Loss Notification (ELN) Deployment 74

5.1 Introduction 74

5.2 Explicit Loss Notification Algorithm Design 75

5.3 Simulation Setup 78

5.4 Simulation Results 79

5.4.1 Wireless Networks with no congestions and no errors 80
5.4.2 Wireless Networks with congestions and no errors 80
5.4.3 Wireless Networks with no congestions and errors 81
5.4.4 Wireless Networks with congestions and errors 81
5.4.5 Comparison of TCP-Reno and TCP-Vegas 85

5.5 Summary 88

5.6 Conclusion 89

Chapter 6. Feedback-Based Congestion And Explicit Window Adaptation In Hybrid Networks With Wired And Wireless Links 90

6.1 Introduction 90

6.2 Hybrid Network Model and Design Architecture 92

6.3 Explicit Window Adaptation Algorithm 94

6.4 Simulation Setup 95

6.5 Simulation Results 96

6.5.1 Performance of FICC 96

6.5.2 Performance of Explicit Window Adaptation Algorithm 101

6.6 Summary 103

Chapter 7. Conclusions and Future Work 104

7.1 Conclusions 104

7.2 Future Work 105

Bibliography 107

Publication 111

List of Figures

Figure 1.1 Comparison of TCP on wired and wireless networks 2
Figure 1.2 Overall research mapping layers 4

Figure 2.1 Internet Protocol Architecture (EITF) 12
Figure 2.2 TCP Header 14
Figure 2.3 Ad hoc mode vs. infrastructure mode 23
Figure 2.4 End-To-End Scheme 26
Figure 2.5 Split Scheme 27
Figure 2.6 Link Layer Scheme 28

Figure 3.1 RD Header structure 31
Figure 3.2 Design principle of FICC 31
Figure 3.3 Congestion control function of FICC 32
Figure 3.4 Description of FICC Algorithm 33
Figure 3.5 Simplified User's View of NS 36
Figure 3.6 C++ and OTcl: The Duality 37
Figure 3.7 Architectural View of NS 38
Figure 3.8 Class Hierarchy (Partial) 39
Figure 3.9 Node (Unicast and Multicast) 40
Figure 3.10 Link 41
Figure 3.11 Inserting Trace Objects 41
Figure 3.12 Monitoring Queue 42
Figure 3.14 NS Packet Format 43
Figure 3.15 NS Directory Structure 44
Figure 3.16 Illustration of a simple hybrid network and WFICC control loop 46

Figure 4.1 FICC on wired network scenario 53
Figure 4.2 Queue length at bottleneck with DropTail 56
Figure 4.3 Queue length at bottleneck with FICC queue 57
Figure 4.4 Congestion window size without FICC 57
Figure 4.5 Congestion window size with FICC 58
Figure 4.6 Throughput without FICC 58
Figure 4.7 Goodput with FICC 59
Figure 4.8 FICC on wired-cum-wireless network scenario 59
Figure 4.9 Queue length without FICC 62
Figure 4.10 Queue length with FICC 62
Figure 4.11 Congestion window without FICC 63
Figure 4.12 Congestion window with FICC 63
Figure 4.13 Goodput without FICC 64
Figure 4.14 Goodput with FICC 64
Figure 4.15 Queue Length without FICC 65
Figure 4.16 Queue Length with FICC 65

Figure 4.17 Window without FICC 66
Figure 4.18 Window with FICC 66
Figure 4.19 Goodput without FICC 67
Figure 4.20 Goodput with FICC 67
Figure 4.21(a,b,c) Queue length and packet losses 69
Figure 4.22 End-to-end delay 69
Figure 4.23 Throughputs and goodputs for different error rates 70
Figure 4.24 Goodputs of different sessions 71

Figure 5.1 Flowchart for TCP packets at Base Station 74
Figure 5.2 Flowchart for ACK packets at Base Station 75
Figure 5.3 Flowchart for ACK packets at Sender 76
Figure 5.4 A simulation model 77
Figure 5.5 Goodput Comparison 78
Figure 5.6 Average Goodput Comparison at different PERs 79
Figure 5.7 Queue length Comparison 80
Figure 5.8 Window Comparison 81
Figure 5.9 Goodputs for different error rates 82
Figure 5.10 Goodput comparison at PER = 10% 82
Figure 5.11 TCP-Reno and TCP-Vegas Comparison at PER = 0% 84
Figure 5.12 TCP-Vegas and TCP-Reno with WFICC at PER = 10% 85

Figure 6.1 A simple hybrid network and the proposed framework 88
Figure 6.2 Traffic Shaper 89
Figure 6.3 Explicit window adaptation algorithm .... 90?
Figure 6.4 A simulation model 90
Figure 6.5 Queue lengths vs. time at bottlenecked link for different error rates 92
Figure 6.6 Queue lengths and packet losses at error rate 1.0% 93
Figure 6.7 Average end-to-end delay 94
Figure 6.8 Throughputs and goodputs for different error rates 94
Figure 6.9 Goodputs of different error rates 96
Figure 6.10 Comparison of TCP window sizes 96
Figure 6.11 Comparison of packet sending rates 97

List of Tables

Table 2.1 Summary of protocols 28

Table 6.1 Minimum, average and maximum queue lengths for different error rates ... 93

Table 6.2 Comparison of throughputs and goodputs for different error rates 95

List of Abbreviations and Terminology

ACK	*Acknowledgement*
ATM	*Asynchronous Transfer Mode*
BS	*Base Station*
CCK	*Complimentary code keying*
cwnd	*Congestion Window*
DCF	*Distributed Coordinator Function*
DiffServ	*Differentiated Services*
DPF	*Down Pressure Factor*
ECN	*Explicit Congestion Notification*
ELN	*Explicit Loss Notification*
EWA	*Explicit Window Adaptation*
FICC	*Fair Intelligent Congestion Control*
FTP	*File Transfer Protocol*
GHz	*Giga-Hertz*
GPLS	*General Packet Radio Service*
GPRS	*General Packet Radio Service*
GSM	*Global*
HTTP	*Hypertext Transfer Protocol*
IntServ	*Integrated Services*
IP	*Internet Protocol*
IPv6	*Internet Protocol version 6*
ISO	*International Standardization Organization*
LAN	*Local Area Network*
MH	*Mobile Host*
OFDM	*Orthogonal Frequency Division Multiplexing*
PCF	*Point Coordinator Function*
PDA	*Personal Digital Assistance*

PER	*Packet Error Rate*
QoS	*Quality of Service*
RF	*Radio Frequency*
SACK	*Selective Acknowledgement*
ssthresh	*Slow Start Threshold*
TCP	*Transmision Control Protocol*
TDMA	*Time Division Multiple Access*
TELNET	*Telecommunications Network*
TOS	*Type of Service*
UDP	*User Datagram Protocol*
URL	*Universal Resource Locator*
WAN	*Wide Area Network*
WFICC	*Wireless Fair Intelligent Congestion Control*
WLAN	*Wireless Local Area Network*
WWW	*World Wide Web*